

BSTZ No. 042390P11056
Express Mail No. EL802874082US

UNITED STATES PATENT APPLICATION

FOR

METHOD AND APPARATUS FOR AVOIDING MULTIPLE PROCESSING OF THE
SAME IPMI SYSTEM EVENT

Inventors

Heung-For Cheng
Mahendra A. Ramachandran
Srinivas Vuppula

Prepared by:

Blakely, Sokoloff, Taylor & Zafman LLP
12400 Wilshire Boulevard, Suite 700
Los Angeles, California 90025
(714) 557-3800

METHOD AND APPARATUS FOR AVOIDING MULTIPLE PROCESSING OF THE
SAME IPMI SYSTEM EVENT

Field

The invention pertains generally to system management software to manage system components. More particularly, the invention relates to a scheme to improve the operation of the Intelligent Platform Management Interface (IPMI) event mechanism so that each event is processed only once.

Background

The Intelligent Platform Management Interface (IPMI), version 1.5, revision 1.0, is an industry initiative for system management software that manages system components such as temperature sensors, voltage sensors, fan sensors, power controls, and other system components and devices. IPMI running on a system, such as a server, may be implemented as a distributed management platform where remote systems may access and manage the IPMI enabled system.

The event mechanism is a major feature in IPMI to indicate the occurrence of a system event to the system management software. The occurrence of an event is recorded in the IPMI System Event Log (SEL) as a SEL record. The management software periodically polls the IPMI SEL records to determine if a new event has been registered. The software may take appropriate actions based on the type of event. For example, if the event is Chassis Intrusion, the software may shutdown the system, and may send a corresponding page or message to the system administrator. In some implementations, it may be necessary for each SEL event to be processed one time only.

Under certain conditions, SEL events may be unnecessarily processed more than once. This is because when the system management software reads a SEL record, the IPMI does not

42390P11056

indicate that the record has been read. This may cause the SEL record to be processed multiple times in some situations.

A few cases in which a SEL record may be unintentionally processed more than once include where 1) a system reboot has occurred, 2) a new operating system is installed in a host system running the system management software, and 3) multiple system management software processes access the SEL to manage the IPMI.

10

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is an illustration showing an exemplary embodiment of a System Event Log including various exemplary record IDs and events.

Figure 2 is a block diagram illustrating a host system running system management software to access the SEL records locally on the IPMI host system.

Figure 3 is a flow diagram illustrating one aspect of the invention to prevent multiple processing of the same SEL record.

Figure 4 shows one embodiment of a system configuration illustrating multiple system management software applications locally managing the same IPMI of a host system.

Figure 5 is a flow diagram illustrating a method of one embodiment of the invention to avoid multiple processing of the same SEL record.

Figure 6 shows one embodiment of a configuration illustrating a host system and multiple remote systems in which the invention may be employed.

Figure 7 shows one embodiment of a second configuration illustrating a host system and a remote system in which the invention may be employed.

Figure 8 is a flow diagram illustrating a method of practicing the invention which applies to any combination of both in-band and out-of-band system management software.

Figure 9 shows one embodiment of a configuration illustrating an embodiment of the invention where a software lock agent synchronizes access to the system event log by one or more out-of-band system management software applications.

Figure 10 shows one embodiment of a second configuration illustrating an embodiment of the invention where a software lock agent synchronizes access to the system event log by both in-band and out-of-band system management software.

DETAILED DESCRIPTION

5 In the following description numerous specific details
are set forth in order to provide a thorough understanding of
the invention. However, one skilled in the art would
recognize that the invention may be practiced without these
specific details. In other instances, well known methods,
procedures, and/or components have not been described in
10 detail so as not to unnecessarily obscure aspects of the
invention.

15 Various aspects of the invention provide novel schemes to
avoid multiple processing of the same SEL record by system
management software. As used herein, system management
software refers to software applications which may process
system events. System management applications may refer to
one or more instances of the same or to different system
management software. A host system may refer to a server,
processing unit, and/or computer unit implementing and running
the IPMI. System management applications can run locally on
20 the host system to control, access, and/or communicate with
the host system through the IPMI. A remote system refers to
any processing unit or computer unit capable of running a
system management application to control, access, and/or
communicate with the host system through the IPMI. References
25 to the Intelligent Platform Management Interface (IPMI) herein
refer to all versions of the IPMI specification and standard
including version 1.5, revision 1.0.

30 Some cases where a SEL record may be unintentionally
processed more than once include where 1) the IPMI host system
is rebooted, 2) a new operating system is installed in a host
system running the system management software, and 3) where
multiple system management software operate on the same IPMI-
enabled host system.

Figure 1 illustrates an exemplary embodiment of a SEL 102 including various exemplary record IDs and events. This may represent a list of SEL records/events as stored by the IPMI of a host system. Note that the events and record IDs shown are for purposes of illustration and a person of ordinary skill in the art would recognize that many other events and record ID schemes may be employed without deviating from the invention.

Figure 2 illustrates an IPMI-enabled host system 202 running system management software locally to monitor and manage the IPMI of the host system 202. This figure illustrates a situation where a SEL record may be processed more than once. For instance, when the host system 202 is rebooted, the system management software application 204 has no way of knowing whether a particular record within the SEL 206 has been previously processed. Thus, it may process the same record(s) again.

Figure 3 is a flow diagram illustrating one embodiment of one aspect of the invention to prevent multiple processing of the same SEL record. To avoid the multiple processing of the same SEL record(s), while the host system is operating the system management software may save the last read SEL record ID in a file in the host system before the host system is rebooted 302. After the next host system reboot 304, during the initialization of the system management software, the record ID in the file is read to determine what is the last read record 306. The records may be arranged in a predetermined order in the SEL event log, for instance in ascending order based on the record ID. The system management software may then use the record ID to request the SEL record 308. The IPMI host system acknowledges the request by returning the requested SEL record along with the next record ID 310. The next record ID is then used to query and start processing the next unprocessed SEL record 312.

However, this mechanism may not solve the problem of multiple processing of SEL records where 1) a new operating system is installed in the host system and 2) multiple system management software applications operate on one or more remote systems to manage the same host system.

Where a new operating system has been installed in the host system, the new system management software running on the new operating system may not be able to access the file that contains the last read SEL record ID in an incompatible file system. Thus, the system management software running on the new operating system may not be able to determine which SEL records have been previously processed by the previous system management software.

Figure 4 illustrates a case where multiple system management software applications 404 and 406 on the host machine 402 may manage the same host system 402. Where multiple system management applications are present, none of the applications may know which SEL records or events have been previously processed or are currently being processed by other system management applications. For example, under the scheme described in Figure 3, multiple system management applications, i.e. 404 and 406, may concurrently process the same event(s) from the system event log 408.

To solve the above problem, one aspect of the invention provides the use of the IPMI non-volatile "Last Software Processed Event ID" storage location to hold the last read SEL record ID. The IPMI Server Management Software (SMS) Message Channel serves as a mutual exclusive mechanism (mutex) for synchronization between multiple system management software reading the SEL.

Figure 5 illustrates one embodiment of this aspect of the invention to avoid multiple processing of the same SEL record. System management software that wants to avoid processing the same SEL record multiple times can request exclusive use of the SEL records by disabling the SMS Message Channel.

Disabling the SMS Message Channel may be accomplished through the IPMI "Enable Message Channel Receive" command 502. The status of the SMS Message Channel determines whether a system management software application can lock and obtain the mutual exclusive use of the SEL.

The IPMI "Enable Message Channel Receive" command returns SUCCESS if the SMS Message Channel was disabled successfully and error if the channel was already disabled 504.

If the SMS Message Channel has already been disabled, the IPMI command returns error status other than SUCCESS 504. This means that another software has disabled the channel to indicate that it is using the SEL. The system management software can then decide to try to disable the SMS Message Channel again 520, exit 522, or wait before retrying to disable the channel again.

If SUCCESS is returned, then the system management software obtains exclusive use of the SEL. The system management software can then obtain the next available unprocessed SEL record and process it.

In one embodiment operating IPMI version 1.5, the command "Get Last Processed Event ID" may be invoked to obtain the last read SEL record ID (LAST_RECORD_ID) 506. From this LAST_RECORD_ID, the system management software can issue an IPMI "Get SEL Entry" command 508. This command returns the last processed record (LAST_RECORD) as well as the next, unprocessed, record ID (NEXT_RECORD_ID).

The system management software may then check the NEXT_RECORD_ID to determine if it is not the END-OF-SEL-RECORD indicator 510. For example, if the returned NEXT_RECORD_ID is "FFFF", this may indicate that no more records are available. If so, the system management software enables the SMS Message Channel and exits.

If the NEXT_RECORD_ID is not the END-OF-SEL-RECORD indicator, the management software then invokes the IPMI "Get

SEL Entry" command to read the next unprocessed record
(NEXT_RECORD) 512.

5 The software issues the "Set Last Processed Event ID"
command with the NEXT_RECORD_ID as parameter to record the
last processed record identification number 514. The
management software then processes this NEXT_RECORD 516.
After the NEXT_RECORD is processed, this completes the cycle
of one SEL record reading and processing.

10 The management software then enables the SMS Message
Channel to release its exclusive use of the SEL 518.

15 In one implementation, the management software then
checks to see if more SEL records are available to be
processed 520. If so, the management software attempts to
again obtain exclusive use of the SEL, via the SMS Message
Channel, and repeat the above process. If no other records
are available for processing, the management software exits
522.

20 In Figure 5, the exemplary mechanism of this aspect of
the invention assumes two things. First, all system
management software that does not want to reprocess the same
SEL record multiple times follows this algorithm. Second, the
SMS Message Channel is only used for the purpose described in
this mechanism.

25 According to one embodiment, the method described and
shown in Figure 5, may only apply to in-band system management
software. In-band software is that which accesses the host
system locally. The method of the invention shown in Figure 5
may apply only where in-band system management software is
employed. This may be because one or more IPMI commands, such
30 as "Enable Message Channel Receive", may not be accessed by
out-of-band software or applications.

35 Out-of-band software on the other hand refers to system
management software on a remote system/machine that does not
rely on an operating system to connect to, communicate with,
and access the IPMI running on a host system/machine. Out-of-

band software does not rely on the operating system on the system/machine hosting the IPMI for connection, communication, and access to the host system/machine but rather may rely on firmware to obtain access to the SEL records.

5 The mechanism illustrated in Figure 5 may not solve the problem of multiple processing of SEL records where 1) both in-band and out-of-band system management software are employed, and 2) only out-of-band system management software runs on one or more remote systems to manage an IPMI host system.

10 Figure 6 illustrates remote systems 604 and 608 running out-of-band system management software which may use dial-up connections 606 and 610 or some network cable connection to access a distributed management platform server (host system) 602 running IPMI.

15 Figure 7 illustrates an IPMI host system 702 which may be accessed by both an in-band and out-of-band system management applications running on a remote 708.

20 In both of the cases illustrated in Figures 6 and 7, SEL records/events may be processed more than once since each system management application does not know whether another system management application has previously processed the same SEL record/event. That is, under certain conditions, the out-of-band system management software may not be able to implement the method illustrated in Figure 5. For instance, the out-of-band system management software running on remote system 604 and 608 may not be capable of remotely invoking the necessary IPMI commands necessary to carry out the method.

25 Figure 8 illustrates another aspect of the invention which applies to any combination of both in-band and out-of-band system management software to avoid processing the same SEL record(s) more than once. This method provides yet another mechanism which provides mutual exclusive access to the SEL.

In one embodiment of the invention, a software process called Software Lock Agent (SLA) is implemented on the host/managed system (the system implementing IPMI). When any system management software or application wants exclusive
5 access, or mutex lock, of the SEL "Last Software Process Event ID" storage location, it sends a "Lock Acquire" request into the Receive Message Queue (RMQ) for the SLA 802.

The SLA then responds to this request 804. If another management application has already requested the lock from the
10 SLA, then the lock is unavailable to subsequent requesting applications and the SLA responds with a "Lock Denial" acknowledgement to the requester 806. The requesting system management application may then try to resend the "Lock Acquire" request to get the lock, retry after a wait period,
15 or exit 824.

If the SLA acknowledges the lock request with a "Lock Acquire OK" acknowledgement, this indicates that no other application presently holds the lock and the sender/requester has mutex lock of the SEL.

The requesting system management application can then access to the "Last Software Process Event ID" storage location, by issuing an IPMI "Get Last Processed Event ID" command or otherwise, to obtain the last read SEL record ID (LAST_RECORD_ID 808). From this LAST_RECORD_ID, the
20 management software can issue an IPMI "Get SEL Entry" command 810. This command returns the last processed record (LAST_RECORD) as well as the next, unprocessed, record ID (NEXT_RECORD_ID).

In one embodiment, the system management software may
30 then check the NEXT_RECORD_ID to determine if it is not the END-OF-SEL-RECORD indicator 812. For example, if the returned NEXT_RECORD_ID is "FFFF", this may indicate that no more records are available. If so, the system management software may send a "Lock Release" request to the SLA and exit.

If the NEXT_RECORD_ID is not the END-OF-SEL-RECORD indicator, the management software then invokes the IPMI "Get SEL Entry" command, using the NEXT_RECORD_ID as a parameter, to read the next unprocessed record (NEXT_RECORD) 814.

5 The software then issues the "Set Last Processed Event ID" command with the NEXT_RECORD_ID as parameter 816. This completes the cycle of one SEL record reading, processing, and marking. The management software then processes this NEXT_RECORD 818.

10 When it finishes, the system management software then sends a "Lock Release" request to the SLA, via the RMQ or otherwise, to release its exclusive use of the SEL 820. The SLA then releases the lock of the SEL so that it can be granted to the next "Lock Acquire" requester.

15 In one implementation, the management software then checks to see if more SEL records are available to be processed 822. If so, the management software attempts to again obtain exclusive use of the SEL, via the SLA lock, and repeat the above process. If no other records are available for processing, the management software exits 824.

20 Since the SLA is a software application, an operating system is assumed to be present and running on the managed system. If the operating system is not running, no in-band system management software runs either. In that case, 25 assuming a single out-of-band management software is running, if an out-of-band management software doesn't receive an acknowledgement from SLA after sending the "Lock Acquire" request 804 for a time-out period, say 30 seconds, it can assume that the operating system is not running and it can 30 exclusively access to the "Last Software Process Event ID" storage location.

35 However, without an operating system running, the method illustrated in Figure 8 cannot support the case where multiple out-of-band management software applications are trying to monitor and control the IPMI on a single host system. That

is, without the operating system the SLA cannot run on the host system to synchronize write access to the "Last Software Process Event ID" storage location.

Figure 9 illustrates how in one embodiment of the invention a software lock agent (SLA) synchronizes access to the system event log by one or more out-of-band system management software applications. The software lock agent may run on a host system 902 and communicate with one or more out-of-band system management software applications (i.e. in remote systems 904 and 908) to control and synchronize access to the system event log. The out-of-band system management software may request exclusive access to the system event log via the software lock agent. If exclusive access to the system event log is not presently assigned to another system management software, then the software lock agent grants exclusive access to the first system management software to make such request. Otherwise, the software lock agent rejects the request. A system management software which has obtained a lock or exclusive access to the SEL may release its lock or exclusive access by sending a message to the software lock agent when it is done processing.

Figure 10 illustrates another embodiment of the invention where a software lock agent synchronizes access to the system event log by one or more in-band system management software applications (i.e. in host system 1002) and one or more out-of-band system management software (i.e. in remote system 1008). The software lock agent illustrated in Figure 10 operates much like the software lock agent illustrated in Figure 9 and described above to control and synchronize exclusive access to the system event log by both in-band and out-of-band system management software.

According to one embodiment, while the software lock agent may coordinate exclusive access to the SEL, it does not prevent access to the SEL per se. That is, system management software applications that ignore the access control mechanism

of the software lock agent may access the SEL despite another system management software having received an exclusive use lock over the SEL from the software lock agent.

While certain exemplary embodiments have been described and shown in the accompanying drawings, it is to be understood that such embodiments are merely illustrative of and not restrictive on the broad invention, and that this invention not be limited to the specific constructions and arrangements shown and described, since various other modifications may occur to those ordinarily skilled in the art. Additionally, it is possible to implement the invention or some of its features in hardware, programmable devices, firmware, software or a combination thereof where the software is provided in a processor readable storage medium such as a magnetic, optical, or semiconductor storage medium.